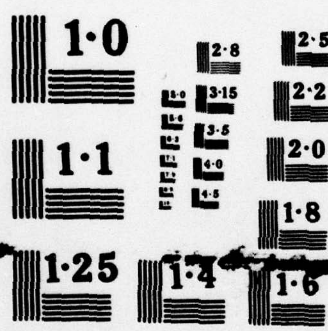
AD-A070 778　ULTRASYSTEMS INC IRVINE CA OPERATIONS RESEARCH AND E--ETC　F/G 5/1
EDATL FACILITY SCHEDULING PROBLEM RESEARCH.(U)
JUN 78　R J EGBERT, J G RAU　N00014-77-C-0190

UNCLASSIFIED　NL

1 OF 1
AD
A070778

END
DATE
FILMED
8-79
DDC


1.0
2.8
2.5
3.15
2.2
3.5
2.0
4.0
4.5
1.1
1.8
1.25
1.4
1.6

(1)

(9) FINAL REPORT ON THE

(6) EDATL FACILITY SCHEDULING

PROBLEM RESEARCH

DDC
JUL 2 1979
C

(10)

BY: R. J. EGBERT ~~AND~~ J. G. RAU
(3) OPERATIONS RESEARCH AND ECONOMIC ANALYSIS DEP~~ARTMENT~~
(1) ULTRASYSTEMS, INC.
(2) IRVINE, CA~~LIFORNIA~~ 92715

(12) 68p.

(11) 1 JUNE 1978

(15)

PREPARED FOR:

NAVAL ANALYSIS PROGRAMS
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-77-C-0190 new
TASK NUMBER NR 274-270

new 411 254

~~JOB~~

TABLE OF CONTENTS

1.0 OVERVIEW OF RESEARCH PROJECT

This is the final report (see Ref. 1—3 for earlier intermediate reports) concerning a research project which focusses on the development of scheduling algorithms and the demonstration of the use of sequencing and scheduling techniques research in the planning and ordering of Naval activities. A specific problem area deals with the scheduling of test activities at Naval test, development and evaluation facilities. For example, at the Naval Ocean Systems Center in San Diego there is presently under construction the Electronics Development and Testing Laboratory (EDATL) which has the mission ~~(see Ref. 4)~~ "to provide facilities and technical support to permit complete integration and simultaneous multi-platform testing of total electronic command control, communications, surveillance, intelligence, and ocean surveillance systems in an electromagnetically shielded and controlled environment." The EDATL will contain a core of ADP equipment available for use by projects on a resource sharing basis.

To fully utilize the EDATL facilities and equipments, scheduling techniques that accurately predict workloads need to be developed and implemented. The biggest problem is in the common area, where many users require similar system suits. Thus, an identification of users requiring a system configuration is a measure of the requirements for processor, disk, tape drive, and other standard ADP type devices. Unfortunately, a piece of equipment may only belong to one system at a time, and there may be periods where people needing EDATL services are effectively frozen out due to unavailability of equipments needed to complete their system configuration. This suggests that perhaps the allocation and prediction of dedicated facility/equipment requirements, and the movement of equipments from dedicated to common areas and vice versa should be scheduled on a priority basis. Proper scheduling will keep track of equipment availabilities projected over time, so that effective, non-wasteful use is made of the facility.

The equipment assets in EDATL can be generally categorized as follows:

(1) common user/shared use - equipment installed for the benefit of all customers who will use EDATL and available for use, on a scheduled basis, to anyone who has a need

(2) <u>customer owned/usable at non-interference times</u> - equipment procured and installed in EDATL by a customer who has agreed to permit other customers the use of the equipment when his own schedule permits

(3) <u>customer-owned/nonsharable</u> - equipment procured and installed in EDATL by a customer who has determined that the equipment will not be available to other users at any time.

Table A-1 in Reference 1 provided an identification of the EDATL equipment assets according to these categories.

All common user equipments are interconnected via a High Speed Data Switch (HSDS) which is capable of instantaneous hardware suit reconfiguration to accommodate demands from different users. The HSDS along with the Shipboard Data Multiplex System (SDMS) are the two principle components of the Signal and Data Distribution System (SADDS), which is a series of electronic interconnect devices which facilitate the configuration of multiple subsystem elements into a total systems complex.

The EDATL scheduling problem is thus concerned with how to schedule the users of both common user equipment and that which is customer-owned but usable at non-interference times, provided it is accessible through the High Speed Data Switch. In other words, the scheduling research only focusses on the use of "sharable" equipment which can be accessed through the HSDS. The scheduling of nonsharable equipment is assumed to be handled by the owners of this equipment.

For all practical purposes the HSDS can simultaneously service more customers than could ever be expected to want to utilize the common user/sharable equipment. Thus, the only constraint imposed on the ability of the HSDS to service customers is the availability of equipment configurations desired together with the necessary supporting personnel.

Customers requesting "sharable" equipment configurations can be basically characterized by five types of data, namely:

(1) user priority

(2) number and type of equipment configurations desired

(3) time span of usage of each equipment configuration desired

(4) desired start date for testing activities

(5) desired completion date for testing activities

In addition, a customer may also have a preference for the order in which he tests his selected configurations. The problem of concern in this research then is to determine how to schedule, or order, these customers according to their requirements so as to meet one or more objective criteria.

Generally speaking, the use of optimized scheduling procedures such as the type discussed in this research report are not used at other test facilities, however automated schemes are used to keep track of resources utilized, downtimes, and the schedule of activities and events. For example, at the Pacific Missile Test Center, Pt. Mugu, the Range Operations Scheduling Office basically uses a semi-automated scheduling system which consists of computer storage of information regarding range and equipment availability as well as the schedule of users and their time allotments. User requests are then manually scheduled based upon computer recall of previous, or planned, commitments of resources. Similarly, at the Air Force Eastern Test Range, Cape Kennedy, a Mechanized Range Scheduling (MRS) System is used, which is basically an information system, operating in a conversational mode, that allows the users to maintain an up-to-the-minute file fully describing the schedule of all tests on the range, including resource downtime and minor support. This system embodies file maintenance, information display, and an indication of conflicts in resource usage between a proposed test and the tests already scheduled. An integral part of this system is the Range (On-Line) Scheduling & Information Exchange (ROSIE) system which maintains historical data that is later formatted into reports on range scheduling by weekly and monthly tests, including monthly downtime. The overall MRS system together with ROSIE is then used as the basis for manual scheduling of range activities. Manual scheduling also appears to be utilized at both Vandenberg AFB, California and at White Sands Missile Range, New Mexico.

Upon contact with the Armament Development and Test Center (ADTC) at Eglin AFB, Florida, it was determined that the range scheduling staff currently use an automated range scheduling program (Ref. 9) which is interactive with a CRT terminal. The purpose of this program is to compare

requested missions with resource availability to produce a mission schedule. An optimization scheme using resource weighting is employed to determine the schedule of mission requests. This particular scheme considers scheduling priorities, the acceptability of alternative resources and the overall time limits in satisfying the maximum number of resource requests while avoiding overtime support wherever possible. Scheduling priorities are based on the ADTC established priority of the project requesting the mission. These priorities are derived directly from the Air Force Precedence Rating of the projects. With regard to the acceptability of alternative resources, a mission that requires a particular resource and cannot use any alternative one would receive scheduling preference over another mission that could either use this resource or some other alternative. Furthremore, the scheduling algorithm employed attempts to place resource requests within the desired time usage interval whenever possible. A mission that requires a particular resource at a specified time of day has a better chance of securing this resource at this time than another mission that could use this resource at any time during the normal work day. This automated range scheduling program is similar in its desired use objectives to the methodology which was developed in this research program, however the method employed is substantially different.

In summary, the basic objectives of this research are to demonstrate the use of sequencing and scheduling techniques in the planning of Naval activities and to show, in particular, how improvements can be obtained in the efficient use of test facilities. The Electronics Development and Testing Laboratory at NOSC has been selected as the "test bed" for the development of these techniques and subsequent implementation. This is a new facility offering a complexity of user requirements and a broad variety of command control and communications system users, hence is a suitable candidate to provide the basis for this research. It is expected that the techniques developed will be sufficiently general that they can be used at other Naval facilities.

2.0 FORMULATION OF THE SCHEDULING PROBLEM

In the case of a Naval test facility, such as EDATL, where there are a multitude of users, each with a different set of test requirements, the scheduling problem mathematical model cannot be expected to be as

easily formulated or solved as in the case of the known, relatively simple cases of job scheduling or even assembly line balancing. In fact, because of the varying resource requirements relative to space, supporting staff, ADP equipment necessary, etc., and constraints such as special security precautions, time deadlines, user priorities, etc., the problem and its corresponding solution given by a scheduling algorithm are considerably more complex than those problems being considered by many researchers today in the area of sequencing and scheduling.

An approach to formulation of the mathematical model for the EDATL scheduling problem would be to use integer variables to characterize the test configurations desired by a user of the sharable equipment. For example, if we define

- N = number of types of possible common user equipments and tenant-owned but available upon permission when schedule permits equipments, i.e., the number of "sharable" equipments
- U = initial number of customers requesting use of sharable equipment
- C_j = number of equipment configurations desired for testing by the $j^{\underline{th}}$ customer
- $T_i^{(j)}$ = amount of time the $i^{\underline{th}}$ equipment configuration is desired for testing by the $j^{\underline{th}}$ customer,

then we could describe each potential user's test requirements by a set of (N+1)-tuples of the form

$$(x_{i1}^{(j)}, x_{i2}^{(j)}, \ldots, x_{iN}^{(j)}; T_i^{(j)})$$

for $i = 1, 2, \ldots, C_j$ where

- $x_{ik}^{(j)}$ = number of items of sharable equipment of the $k^{\underline{th}}$ type desired in the $i^{\underline{th}}$ test configuration by the $j^{\underline{th}}$ user.

In other words, one could use a set of vectors whose first N coordinates consist of non-negative integers to describe the configuration needed. A coordinate equal to 0 would mean that the equipment represented by this coordinate was not needed, whereas a non-zero coordinate would mean that one or more of this type of equipment is needed.

For example, if a customer's test configuration requirements were given by (0, 1, 1; 3) and (2, 0, 1; 2), then this would mean that he desires to test two equipment configurations consisting of using one item each of equipments #2 and #3 for a total of 3 time units in the first configuration and of using two items of equipment #1 and one item of equipment #3 for a total of 2 time units in the second configuration.

An additional consideration would be the specification by a customer of the desired time at which he would like to test a particular configuration or the interval of time over which he would like to conduct all of his testing. This feature could be accounted for by defining

$S_i^{(j)}$ = desired time at which the $j^{\underline{th}}$ customer would like to test his $i^{\underline{th}}$ equipment configuration

$E_i^{(j)}$ = desired time at which the $j^{\underline{th}}$ customer would like to have completed the test of his $i^{\underline{th}}$ equipment configuration.

For the case of a customer with a firm requirement to start testing at a particular time, then one would define $E_i^{(j)} = S_i^{(j)} + T_i^{(j)}$. On the other hand, if a customer desired to conduct the testing of a particular configuration over the interval [A,B] and complete testing by time B, then one would define $S_i^{(j)} \geq A$ and $E_i^{(j)} = B - T_i^{(j)}$. To incorporate this type of consideration in the above characterization of test requests, test requirements could be described by a set of (N + 3)-tuples of the form

$$(x_{i1}^{(j)}, x_{i2}^{(j)}, \ldots, x_{iN}^{(j)}; T_i^{(j)}, S_i^{(j)}, E_i^{(j)})$$

for $i = 1, \ldots, C_j$ and $j = 1, \ldots, U$.

In terms of plausible criteria for the scheduling of users of sharable equipment there are several options, namely:

(1) maximize the total revenue to be derived from common user equipment usage

<u>COMMENT</u>

This objective criterion is based on the premise that common user equipment is EDATL-owned and thus must be paid for out of operating expenses. The charge for equipment which is customer-owned would be assumed to correspond to that which

he pays to EDATL and so by "leasing" his equipment he can reduce his total occupancy charge. Since such "lease fees" would represent a pass-through and EDATL would derive no profit from this sharing arrangement, one would only consider the revenue from usage of that equipment which is EDATL-owned.

(2) maximize the total annual usage (in hours) of all common user equipment

COMMENT

This objective criterion implicitly suggests that maximizing equipment usage is good and, if the cost per hour of usage were the same for each item of common user equipment, then this criterion would be the same as (1). On the other hand, the latter situation may not be the case because of the initial allocation of costs on the basis of space usage and then the proration of these costs on the basis of an annual number of hours of operation. This criterion is consistent with the desire to fully utilize the EDATL assets as much as possible.

(3) minimize the total annual idle time of all common user equipment

COMMENT

This objective criterion is equivalent to (2).

(4) maximize the chances of higher priority requests being scheduled

COMMENT

This objective criterion is applicable to the situation where either users or specific test requests have higher priority than other users and/or test requests, however it is not necessarily possible to schedule all test requests.

An objective function which would allow consideration of any one of the above criteria is given by

$$\sum_{j=1}^{U} \sum_{i=1}^{C_j} f_{ij}\, y_{ij} \tag{2.1}$$

where

$$y_{ij} = \begin{cases} 1 & \text{if the } i^{\underline{th}} \text{ test configuration of the } j^{\underline{th}} \text{ customer is scheduled} \\ 0 & \text{otherwise} \end{cases}$$

f_{ij} = "value" of scheduling the $i^{\underline{th}}$ test configuration of the $j^{\underline{th}}$ customer

For criterion (1),

$$(2.2) \qquad f_{ij} = \sum_{k=1}^{N} R_k \, x_{ik}^{(j)} \, T_i^{(j)}$$

where

R_k = cost per unit time for use of the $k^{\underline{th}}$ type of equipment;

for criteria (2) and (3),

$$(2.3) \qquad f_{ij} = \sum_{k=1}^{N} x_{ik}^{(j)} \, T_i^{(j)} \quad ;$$

for criterion (4),

(2.4) $\quad f_{ij}$ = priority of the $i^{\underline{th}}$ test configuration of the $j^{\underline{th}}$ customer

where "high" values of f_{ij} denote higher priority. (NOTE: In this case the objective is to maximize the sum of the priorities, which would imply that higher priority items have a greater chance of being scheduled.)

The problem then of interest is to find a schedule over a given time interval, say [0,T], which maximizes $\sum_{j=1}^{U} \sum_{i=1}^{C_j} f_{ij} \, y_{ij}$ over all $y_{ij} = 0$ or 1 and subject to equipment availability constraints. An example of a constraint of the latter type would be if there are N_k items of equipment of type k and the total time the $v^{\underline{th}}$ item of equipment of type k is unavailable in [0,T] is T_{vk}, then a family of equipment availability constraints is given by

$$(2.5) \qquad \sum_{j=1}^{U} \sum_{i=1}^{C_j} x_{ik}^{(j)} \, T_i^{(j)} \, y_{ij} \le \sum_{v=1}^{N_k} (T - T_{vk}) = N_k \, T - \sum_{v=1}^{N_k} T_{vk}$$

for $k = 1, \ldots, N$. This type of constraint is based on the observation that the total usage hours cannot exceed the total hours available for use.

Equipment availability constraints must include consideration of whether or not a particular piece of equipment of the type desired is available for the time interval needed. A method for characterizing equipment availability is to define equipment availability functions of the form

$$A_{jk}(x) = \begin{cases} 1 & \text{if the } j^{\underline{th}} \text{ equipment of type } k \text{ is available} \\ & \text{for use at time } x \\ 0 & \text{otherwise} \end{cases}$$

for $k = 1, \ldots, N$ and $j = 1, \ldots, N_k$. Thus, if a customer desired to test a particular configuration for T time units at time t_1 and required one equipment of type k, then this would be possible if at least one j could be found for which $A_{jk}(x) = 1$ for $x = t_1, t_1 + 1, \ldots, T - 1$.

The use of equipment availability functions enables the formulation of constraints that are time dependent in the following sense. Consider a test configuration of the form $(x_1, x_2, \ldots, x_n; T, t, t + T)$, which represents a requirement to simultaneously test x_k equipments of type k for T time units starting at time t. Define

$$z_{jk} = \begin{cases} 1 & \text{if the } j^{\underline{th}} \text{ equipment of type } k \text{ is used in} \\ & \text{the test} \\ 0 & \text{otherwise} \end{cases}$$

for $k = 1, \ldots, N$ and $j = 1, \ldots, N_k$. Then it would only be possible to schedule this particular configuration for testing at time t if the following conditions were satisfied:

<u>Condition A</u> $$\sum_{j=1}^{N_k} z_{jk} = x_k$$

<u>Condition B</u> $$z_{jk} \left(\sum_{x=t}^{t+T-1} A_{jk}(x) - T \right) = 0 .$$

Condition A states that the number of equipments of type k assigned to the test request must equal the number required, namely, x_k. Condition B states that if the $j^{\underline{th}}$ equipment of type k is assigned then it must be initially available at time t and then must be available for a total of T time units thereafter; that is, $A_{jk}(t) = A_{jk}(t+1) = \ldots = A_{jk}(t+T-1) = 1$.

Since it is reasonable to expect that users of sharable equipment in the EDATL facility will generally desire to conduct tests at either specified times (such as Mondays at 0700) or within a specified time period (such as Tuesdays between 0800 and 1300), a scheduling problem formulation which more readily recognizes this type of consideration and which can be easily formulated as a 0-1 linear programming problem is the following:

$$(2.6) \qquad \text{maximize} \sum_{i \in ELIG(t)} f_i(t)\, y_i$$

subject to

$$(2.7) \qquad \sum_{j=1}^{N_k} z_{ijk} = x_{ik}\, y_i \qquad \text{for } i \text{ in } ELIG(t) \text{ and } k=1, \ldots, N$$

$$(2.8) \qquad z_{ijk}\left(\sum_{x=t}^{t+T_i-1} A_{jk}(x) - T_i \right) = 0 \qquad \text{for } i \text{ in } ELIG(t),\ j=1, \ldots, N_k \text{ and } k=1, \ldots, N$$

$$(2.9) \qquad \sum_{i \in ELIG(t)} z_{ijk} \leq A_{jk}(t) \qquad \text{for } j=1, \ldots, N_k \text{ and } k=1, \ldots, N$$

$$(2.10) \qquad z_{ijk},\ y_i = 0 \text{ or } 1 \qquad \text{for } i \text{ in } ELIG(t),\ j=1, \ldots, N_k \text{ and } k=1, \ldots, N$$

In this problem formulation, the objective is to determine the "best" test configurations to schedule at time t out of all those which are eligible to be tested at time t, as denoted by the set ELIG(t). The $i^{\underline{th}}$ test configuration in ELIG(t) has the form

$$(x_{i1}, x_{i2}, \ldots, x_{iN}; T_i, S_i, E_i)$$

where $S_i = t$, or, if the customer is indifferent to the start time, then $S_i \geq t$ provided that $t + T_i \leq E_i$. We also define the 0-1 variables in this formulation as follows:

$$z_{ijk} = \begin{cases} 1 & \text{if the } j^{\underline{th}} \text{ equipment of type } k \text{ is assigned to the } i^{\underline{th}} \text{ configuration} \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if the } i^{\underline{th}} \text{ configuration is scheduled} \\ 0 & \text{otherwise} \end{cases}$$

Constraints (2.7) and (2.8) are conditions A and B, respectively, as defined previously. Constraint (2.9) is necessary in order to prevent the same piece of equipment from being assigned to more than one test configuration simultaneously. The function $f_i(t)$ is chosen to be time dependent to allow different criteria to be employed at different times. An example of the use of this feature would be to increase the priority of a test configuration which was eligible to be scheduled earlier, but was not selected as part of the optimal solution at an earlier time.

The preceding formulation is to be regarded as the formulation of the EDATL scheduling problem. In Section 4.0 a procedure is discussed for the solution of this problem and an example is presented.

3.0 SCHEDULABILITY OF TEST CONFIGURATIONS

The problem given by (2.6)—(2.10) is potentially one of large dimension since, if there are N_k equipments of type k, N equipment types, C_j test requests by the $j^{\underline{th}}$ customer, and U customers, then

(3.1) $$\text{maximum number of } z_{ijk}\text{'s} = \left(\sum_{k=1}^{N} N_k\right)(N)\left(\sum_{j=1}^{U} C_j\right)(U)$$

and

$$\text{maximum number of } y_i\text{'s} = \left(\sum_{j=1}^{U} C_j\right)(U) \; .$$

Therefore, at worst case, the dimensionality of the problem is given by

$$\left(\sum_{j=1}^{U} C_j\right)(U)\left[\left(\sum_{k=1}^{N} N_k\right)(N) + 1\right] .$$

This discussion suggests that, wherever possible, steps should be taken to reduce the dimensionality of the problem.

The determination of whether or not the elements in a set of test configurations can all be scheduled within the allotted time span is complicated by the presence of "free" configurations. A free configuration is a test configuration which can be scheduled independently of any other configuration in a given set of test configurations. For example, in the set of configurations given by

$$\mathcal{A} = \{(1, 0, 0; 1), (0, 1, 0; 2), (0, 0, 1; 3)\}$$

all configurations are free since they have no equipment requirements in common, whereas in the set of configurations given by

$$\mathcal{B} = \{(1, 0, 0; 1), (1, 1, 0; 2), (0, 0, 1; 3)\}$$

only (0, 0, 1; 3) is a free configuration. Because of the potentially large number of possible orderings of test configurations to be considered in determining which ones would be scheduled, it is desirable to remove the free configurations from consideration thus reducing the number of combinations which need to be considered.

A technique for accomplishing this can be described as follows. Suppose we are given the following eligible test configurations at a given point in time:

$$(x_{11}, x_{12}, \ldots, x_{1N}; T_1)$$
$$(x_{21}, x_{22}, \ldots, x_{2N}; T_2)$$
$$\vdots$$
$$(x_{C1}, x_{C2}, \ldots, x_{CN}; T_C),$$

where

x_{ik} = *number of items of sharable equipment of the* $k^{\underline{th}}$ *type desired in the* $i^{\underline{th}}$ *test configuration*

T_i = *amount of time desired for testing the* $i^{\underline{th}}$ *equipment configuration*

Let N_k = number of items of equipment type k available and assume that all items of equipment are available for use during the time period of interest (i.e., the equipment availability functions are all equal to 1.0). Now, define

$$(3.3) \qquad e_k = \begin{cases} 0 & \text{if } N_k - \sum_{i=1}^{C} x_{ik} \geq 0 \\ 1 & \text{otherwise} \end{cases}$$

for $k = 1, \ldots, N$, and

$$(3.4) \qquad F(i) = \begin{cases} 1 & \text{if } \sum_{k=1}^{N} x_{ik} e_k = 0 \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, C$. We assert that if $F(i) = 1$ then the $i^{\underline{th}}$ test configuration is free, and if $F(i) = 0$ then the $i^{\underline{th}}$ test configuration is not free. This result follows from the observation that $N_k - \sum_{i=1}^{C} x_{ik}$ denotes the total possible demand for equipments of type k; thus if $N_k - \sum_{i=1}^{C} x_{ik} \geq 0$, then the demand for equipments of type k can be met simultaneously with no waiting. Whenever the equipment requirements of a given test configuration are such that in conjunction with the corresponding equipment requirements of the other configurations under consideration all requirements can be met (i.e., for each choice of k we have $N_k - \sum_{i=1}^{C} x_{ik} \geq 0$), then this configuration would be free. This will occur if, and only if, $F(i) = 1$ where i corresponds to the configuration being examined. In the preceding examples, assuming only one equipment of each type, for the set $\mathcal{A}$ we have $e_1 = e_2 = e_3 = 0$, thus all configurations are free, whereas for the set $\mathcal{B}$ we have $e_1 = 1$ and $e_2 = e_3 = 0$ which implies that only (0, 1, 1; 3) is free. If we changed $N_1 = 1$ to $N_1 = 2$, then all elements of $\mathcal{B}$ would be free.

Therefore, free configurations are ones which can be automatically scheduled and thus removed from consideration in solving the optimization problem given by (2.6)–(2.10). This would then reduce the dimensionality of the problem and thus would facilitate easier computation of the optimal schedule.

Using the concept of freeness, a simple test for schedulability of a given set of test configurations can be derived as follows, under the assumption that all equipments are initially available throughout the entire test period [0,T]. Consider C configurations where the $i^{\underline{th}}$ test configuration has the form $(x_{i1}, x_{i2}, \ldots, x_{iN}; T_i)$ and define "pairwise freeness" between two distinct configurations i and j (i.e., we say that i is free relative to configuration j) to mean that $x_{ik} + x_{jk} \le N_k$ for all k = 1, ..., N. Now, let

$$F_{ij} = \begin{cases} 0 & \text{if configuration } i \text{ is free relative to configuration } j \\ 1 & \text{otherwise} \end{cases}$$

$(ST)_i$ = start time for testing of the $i^{\underline{th}}$ configuration

$(EL)_i$ = elapsed calendar time until completion of testing of the $i^{\underline{th}}$ configuration.

If the configurations are considered for testing in the order 1, 2, ..., C, then it is easily verified that

$$(3.5) \qquad (ST)_i = \max_{j<i} F_{ij}\,(EL)_j$$

and the total elapsed calendar time through testing of the first i configurations is given by

$$(3.6) \qquad (TEL)_{1,2,..,i} = \max_{j=1,..,i} (EL)_j = \max_{j=1,..,i} \left\{(ST)_j + T_j\right\}.$$

Therefore, the number of those configurations which can be tested in [0, T] is the largest value of i, say i*, such that

$$(3.7) \qquad (TEL)_{1,2,..,i^*} \le T \text{ and } (TEL)_{1,2,..,i^*+1} > T .$$

If $(TEL)_{1,2,...,C} \le T$, then all configurations can be scheduled within [0, T].

As an illustrative example, let us consider the following three test configurations:

Configuration 1: (1, 1, 0, 0; 2)
Configuration 2: (1, 1, 1, 0; 4)
Configuration 3: (0, 0, 1, 2; 3),

where we are assuming four types of equipment, all initially available throughout the test period, there is only one item of equipment of types 1, 2 and 3, and there are two equipments of type 4. Performing the pairwise test for "freeness", the F_{ij}'s can be represented by the following 3x3 matrix:

$$(3.8) \qquad F = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} ,$$

since $F_{ii} = 1$ for i = 1, 2, 3. If we consider the test interval [0, 7] and apply this previously described test for schedulability, then we obtain the following results:

Candidate Schedule Order	Schedulable Sub-Order	Total Elapsed Time
1, 2, 3	1, 2	6
1, 3, 2	1, 3, 2	7
2, 1, 3	2, 1	6
2, 3, 1	2, 3	7
3, 1, 2	3, 1, 2	7
3, 2, 1	3, 2	7

Since both configurations 1 and 3 are free of each other, it follows that both configurations can be scheduled at time 0 and then configuration 2 can be scheduled at time 3 which is the completion time for testing of configuration 3.

Under the assumption that all items of equipment are initially available in the time period [0,T], then a simple necessary condition for whether or not the given set $(x_{11}, x_{12}, \ldots, x_{1N}; T_1)$, $(x_{21}, x_{22}, \ldots, x_{2N}; T_2)$, ..., $(x_{C1}, x_{C2}, \ldots, x_{CN}; T_C)$ of C test configurations can all be scheduled in the interval [0,T] is as follows. If all C configurations are schedulable, then it must be true that

$$(3.9) \qquad T \geq \max_{k=1,\ldots,N} \frac{1}{N_k} \sum_{i=1}^{C} x_{ik} T_i \quad .$$

Inequality (3.9) follows from the observation that the total usage time for equipments of type k, given by $\sum_{i=1}^{C} x_{ik} T_i$, must not exceed $N_k T$ for each k, i.e., we must have

$$(3.10) \qquad T \geq \frac{1}{N_k} \sum_{i=1}^{C} x_{ik} T_i$$

for k = 1, ..., N; hence the inequality follows.

As an illustration, consider the case of three equipment types where $N_1 = 1$, $N_2 = 3$ and $N_3 = 2$. Figure 3.1 shows that the following six test configurations can be scheduled in a minimum of 8 time units:

Number	Configuration
1	(1,1,0;2)
2	(0,1,1;4)
3	(1,2,0;3)
4	(0,1,1;2)
5	(0,3,0;1)
6	(1,1,1;2)

It is readily observed that

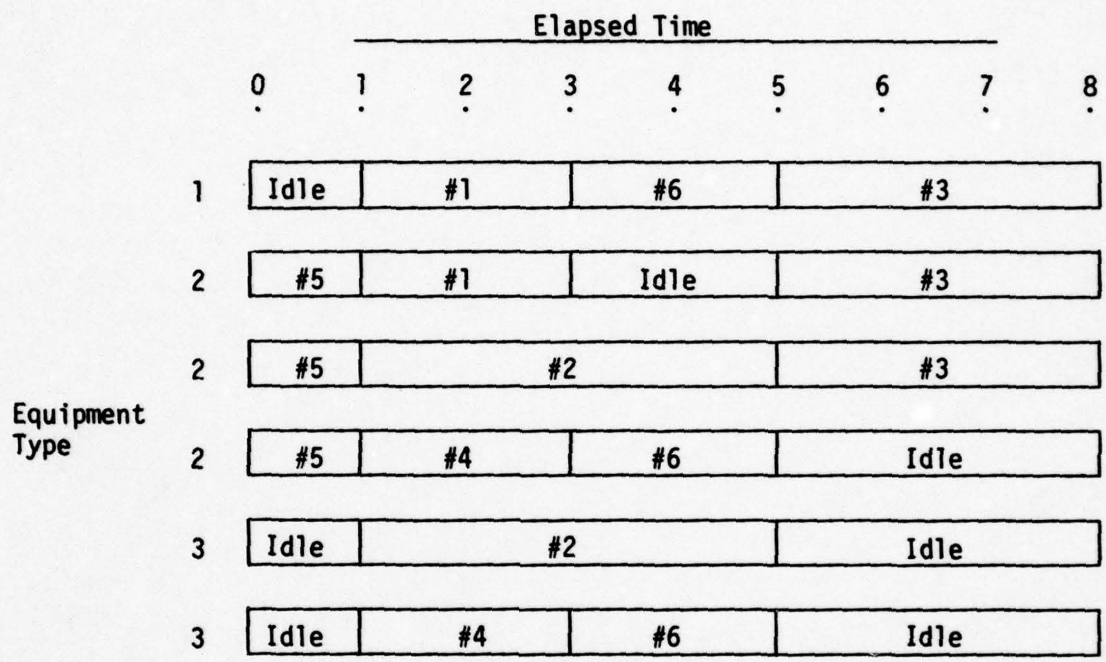
$$(3.11) \qquad \frac{1}{N_k} \sum_{i=1}^{C} x_{ik} T_i = \begin{cases} 7 & \text{for } k = 1 \\ \frac{19}{3} & \text{for } k = 2 \\ 4 & \text{for } k = 3 \end{cases} ;$$

hence

$$(3.12) \qquad \max \left\{ 7, \frac{19}{3}, 4 \right\} = 7 \leq 8 \quad .$$

In terms of sufficient conditions for schedulability the obvious condition is that $\sum_{i=1}^{C} T_i \leq T$ for C given test configurations. The formulation of other, less obvious conditions generally requires the development of a scheme which generates a schedule. Thus, if through the use of the

Figure 3.1 Illustrative Example of Least Time Schedule



scheme a given set of test configurations can be scheduled within the desired time period of interest, then the conditions of the scheme, in essence, become the corresponding sufficient conditions. Several such schemes for accomplishing this are: (1) the schedule-according-to-decreasing-time-requirements method and (2) the schedule-as-soon-as-possible method. Each of these methods will be described in the following paragraphs.

Consider C test configurations of the following form: $(x_{11}, x_{12}, \ldots, x_{1N}; T_1), (x_{21}, x_{22}, \ldots, x_{2N}; T_2), \ldots, (x_{C1}, x_{C2}, \ldots, x_{CN}; T_C)$, where there are no specified start or completion times for each of the test requests. The schedule-according-to-decreasing-time-requirements method requires that these configurations be ordered so that $T_1 \geq T_2 \geq \ldots . \geq T_C$. Then configurations are scheduled as soon as possible according to their time requirements. The steps in this procedure are as follows:

(1) Set $n = 1$ and $j_o = 0$.

(2) Let j_n = largest integer x such that $\sum_{i=j_{n-1}+1}^{x} x_{ik} \leq N_k$
for all $k = 1, \ldots, N$.

(3) Set $L_n = T_{j_{n-1}+1}$.

(4) Test $C-j_n$. If ≤ 0, go to (6); if >0, go to (5).

(5) Replace n by n+1, and go to (2).

(6) Compute $T_{max} = \sum_{i=1}^{j_n} L_i$.

(7) If $T_{max} \leq T$, then the C requests are schedulable in [0,T].

For example, consider the case of four equipment types with $N_1 = 1$, $N_2 = 2$, $N_3 = 3$ and $N_4 = 4$, and the following test requests:

Number	Configuration
1	(0,0,1,0;6)
2	(1,0,0,1;5)
3	(0,2,0,0;4)
4	(0,0,2,3;4)

Number	Configuration
5	(1,2,0,1;3)
6	(0,0,0,3;3)
7	(0,0,2,3;2)
8	(0,0,3,0;2)
9	(1,1,2,2;1)

Applying the first six steps of this procedure, it is easily observed that:

$$\begin{array}{lll} j_1 = 4 & \quad & L_1 = 6 \\ j_2 = 6 & & L_2 = 3 \\ j_3 = 7 & & L_3 = 2 \\ j_4 = 8 & & L_4 = 2 \\ j_5 = 9 & & L_5 = 1 \end{array}$$

thus, from step (6),

$$T_{max} = \sum_{i=1}^{j_5} L_i = 14 \quad . \tag{3.13}$$

Therefore, steps (1)—(7) provide a sufficiency test for whether the nine given test configurations can be scheduled in an interval of at most 14 time units.

The procedure described by steps (1)—(7) is basically one in which test configurations are scheduled in groups. In the preceding example, configurations 1—4 are schedulable in the first 6 time units, configurations 5 and 6 are schedulable in the next 3 time units, configuration 7 is schedulable in the next 2 time units, configuration 8 is schedulable in the next 2 time units, and configuration 9 is schedulable in the next 1 time unit; hence it is possible to schedule all configurations within 14 time units. Of course, it is not known whether or not these test configurations could be scheduled more efficiently in terms of less than 14 time units. A *potentially* more efficient method for "grouping" the test configurations can be described as follows:

(1) Set $S = 1$, $j = 1$ and $i = 1$.

(2) Put configuration i in the schedulable group given by I_j.

(3) Replace i by $i + 1$.

(4) Test $C - i$. If ≥ 0, go to (5); if < 0, go to (9).

(5) Compute $x_{ik} + \sum_{v \in I_j} x_{vk}$. If $\leq N_k$ for all $k = 1, \ldots, N$, then put configuration i in the set I_j and go to (3). If $> N_k$ for at least one $k = 1, \ldots, N$, then go to (6).

(6) Replace j by $j + 1$.

(7) Test $S - j$. If ≥ 0, go to (4); if < 0, go to (8).

(8) Replace S by $S + 1$, put configuration i in the set I_S, set $j = 1$ and go to (3).

(9) Compute

$$T_{max} = \sum_{j=1}^{S} \max_{i \in I_j} T_i \ .$$

(10) If $T_{max} \leq T$, then the C requests are schedulable in $[0,T]$.

To illustrate this revised procedure for the schedule-according-to-decreasing-time-requirements method, consider the preceding example with nine test configurations. In this case, applying steps (1)–(8), it is easily shown that

$$I_1 = \{1, 2, 3, 4\}$$
$$I_2 = \{5, 6, 8\}$$
$$I_3 = \{7\}$$
$$I_4 = \{9\}$$

thus, from step (9),

$$T_{max} = \sum_{j=1}^{4} \max_{i \in I_j} T_i \tag{3.14}$$

$$= 6 + 3 + 2 + 1 = 12 \ . \tag{3.15}$$

This implies that, by "regrouping", the nine test configurations can actually be scheduled in an interval of at most 12 time units. For arbitrary T,

steps (1) – (10) thus provide a sufficiency test for whether or not a given set of C test configurations can be scheduled in the interval [0,T].

The schedule-as-soon-as-possible method is based on the idea that given an ordering of test configurations for consideration then starting with the first configuration in the ordering each other configuration is scheduled as soon as possible based on its relative freeness to other test configurations. To illustrate this procedure, suppose we are given C test configurations of the form $(x_{11}, x_{12}, \ldots, x_{1N}; T_1)$, $(x_{21}, x_{22}, \ldots, x_{2N}; T_2)$, $\ldots$, $(x_{C1}, x_{C2}, \ldots, x_{CN}; T_C)$, where there are no specified start or completion times. This method involves the following steps:

(1) Set $j = 1$ and $i = 1$.

(2) Define $t_j = 0$ and $R_j = \emptyset$.

(3) Put configuration i in the set I_j.

(4) Replace i by $i + 1$.

(5) Test $C - i$. If ≥ 0, go to (6); if < 0, go to (7).

(6) Compute $x_{ik} + \sum_{v \in A_j} x_{vk}$, where $A_j = \bigcup_{k=1}^{j} I_k - \bigcup_{k=1}^{j} R_j$. If $\leq N_k$ for all $k = 1, \ldots, N$, then put configuration i in I_j, set $(ET)_i = T_i + t_j$ and go to (4). If $> N_k$ for at least one $k = 1, \ldots, N$, then go to (4).

(7) Test $\{1, 2, \ldots, C\} - \bigcup_{k=1}^{j} I_k$. If empty, go to (13); if non-empty, go to (8).

(8) Replace j by $j + 1$.

(9) Define $t_j = \min \left\{ (ET)_i : (ET)_i > t_{j-1} \text{ and } i \in \bigcup_{k=1}^{j-1} I_k \right\}$

(10) Define $R_j = \left\{ i : i \in \bigcup_{k=1}^{j-1} I_k \text{ and } t_j \geq (ET)_i \right\}$

(11) Define $x_j = \min \left\{ i : i \notin \bigcup_{k=1}^{j-1} I_k \right\}$

(12) Replace i by x_j and go to (6).

(13) Compute $T_{max} = \max_i (ET)_i$

(14) If $T_{max} \leq T$, then the C requests are schedulable in [0,T].

To illustrate the schedule-as-soon-as-possible method, consider the previous example of nine test configurations with four equipment types such that $N_1 = 1$, $N_2 = 2$, $N_3 = 3$ and $N_4 = 4$. Referring to Figure 3.2, it is readily observed that, applying the steps of this procedure to the ordering of configurations given by 1, 2, ..., 9, the following results are obtained:

$t_1 = 0 \qquad I_1 = \{1, 2, 3, 4\}$

$t_2 = 4 \qquad I_2 = \{6\}$

$t_3 = 5 \qquad I_3 = \{5\}$

$t_4 = 6 \qquad I_4 = \{8\}$

$t_5 = 7 \qquad I_5 = \emptyset$

$t_6 = 8 \qquad I_6 = \{7\}$

$t_7 = 10 \qquad I_7 = \{9\}$

$T_{max} = 11$.

This implies that these nine configurations can be scheduled in an interval of at most 11 time units. For arbitrary T, steps (1)–(13) thus provide a sufficiency test for whether or not a given set of C test configurations can be scheduled in the interval [0,T].

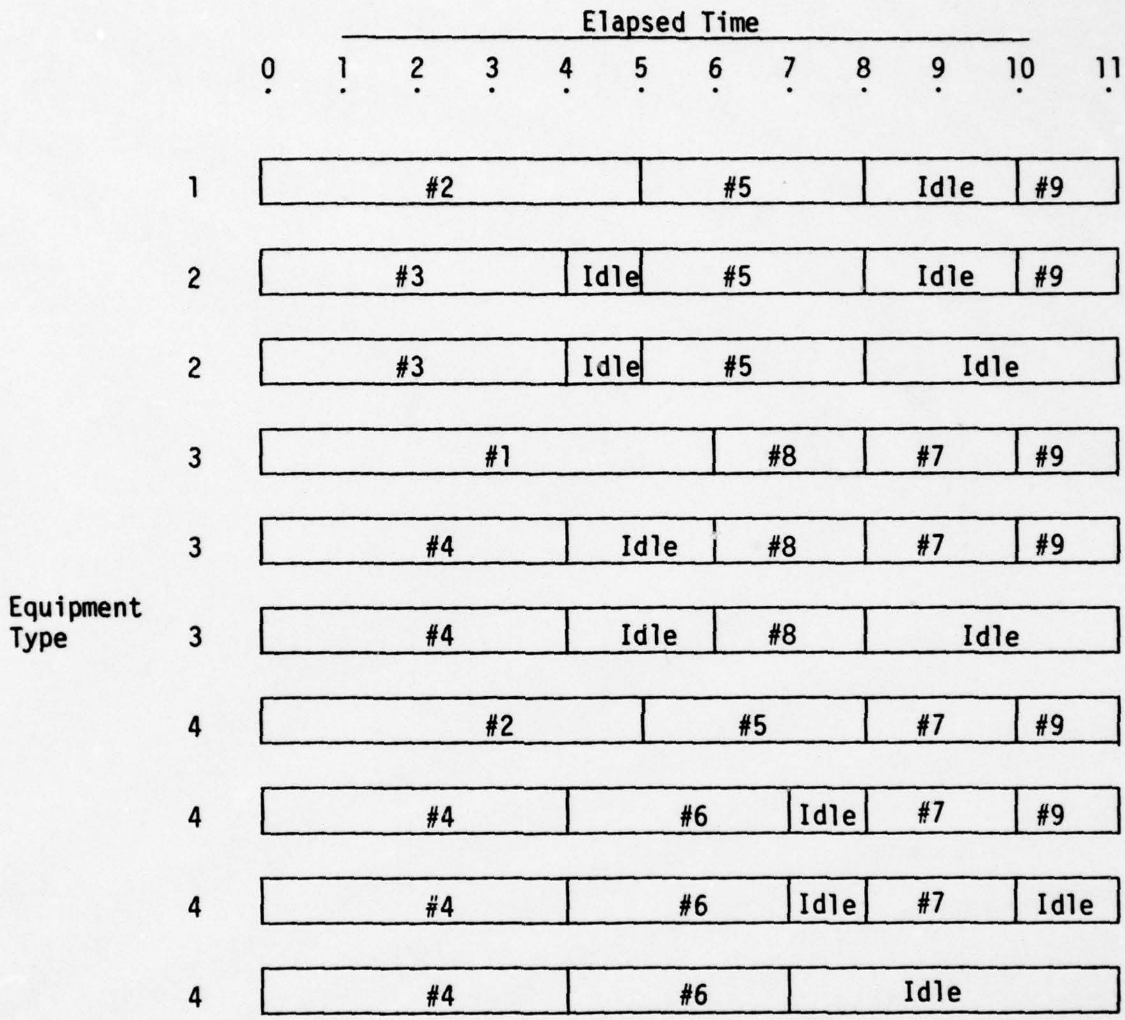
It is important to recognize that the schedule-as-soon-as-possible method presupposes a given ordering of test configurations. As a result, the value T_{max} could be expected to differ depending upon the presupposed ordering. For example, if the ordering assumed had been 5, 9, 8, 6, 7, 4, 3, 2, 1, then it can be shown that the following results would have been obtained:

$t_1 = 0 \qquad I_1 = \{5, 6, 8\}$

$t_2 = 2 \qquad I_2 = \{1\}$

$t_3 = 3 \qquad I_3 = \{9\}$

Figure 3.2 Illustrative Example of the Schedule-As-Soon-As-Possible Method



$t_4 = 4 \quad I_4 = \{2, 3, 7\}$

$t_5 = 6 \quad I_5 = \{4\}$

$T_{max} = 10$

Thus, these nine configurations can actually be scheduled in an interval of at most 10 time units.

Appendix A presents a computer listing (in Fortran IV) which provides for using either the schedule-according-to-decreasing-time-requirements method or the schedule-as-soon-as-possible method. The latter method requires starting with a given ordering. In this regard, the computer program allows for two types of ordering, namely: (1) for a given set of test requests, they are ordered according to decreasing times and then the method is applied, or (2) for a given set of test requests, they are considered in the order in which they are given (i.e., assumed to be analogous to the first-come first-served type of criterion). Appendix B presents illustrative examples derived through the use of this computer program.

Another type of schedulability question deals with the determination of whether or not a given test configuration can be scheduled within an already established schedule of test configurations. This type of situation would arise when the scheduling problem defined by equations (2.6) – (2.10) has been solved, a schedule has been established via the solution, and a new test request is received. The problem then is to determine whether or not this new request can be scheduled within the time period, say [0,T], of the established schedule or must be scheduled at time T or later. For example, suppose this new test request has the form $(x_1, x_2, \ldots, x_N; T_1)$ and we desire to determine if this request can be scheduled at time S, which could represent either the customer's desired time for testing or an arbitrary time in the scheduling period [0,T]. The steps in determining the schedulability of this request can be described as follows:

(1) Set $k = 1$.

(2) Test x_k. If=0, go to (3); if $\neq 0$, go to (5).

(3) Replace k by k+1.

(4) Test N-k. If ≥ 0, go to (2); if < 0, go to (14).

(5) Compute $\sum_{j=1}^{N_k} A_{jk}(S) - x_k$. If <0, go to (13); if >0, go to (6).

(6) Let $J_k = \{j : A_{jk}(S) = 1\}$

(7) Define $C_k = \binom{\text{Card } J_k}{x_k}$

(8) Divide J_k into C_k distinct subsets $I_1, \ldots, I_{C_k}$ each of cardinality x_k so that $J_k = \bigcup_{i=1}^{C_k} I_i$.

(9) Set i = 1.

(10) Compute $\sum_{t=S}^{S+T_1-1} A_{vk}(t) - T_1$ for all $v \epsilon I_i$. If = 0 for all $v \epsilon I_i$, then the configuration is schedulable relative to the $k^{\underline{th}}$ type of equipment so go to (3). If <0 for any $v \epsilon I_i$, go to (11).

(11) Replace i by i+1.

(12) Test $C_k - i$. If ≥0, go to (10); if <0, then the configuration is not schedulable relative to the $k^{\underline{th}}$ type of equipment so go to (13).

(13) Configuration is not schedulable at time S.

(14) Configuration is schedulable at time S.

To illustrate this procedure, consider the situation in which we have five equipment types (N = 5), two equipments of type 1 ($N_1 = 2$), three equipments of type 2 ($N_2 = 3$), one equipment of type 3 ($N_3 = 1$), three equipments of type 4 ($N_4 = 3$), and two equipments of type 5 ($N_5 = 2$). Assume that over the time interval [0,18] the equipment availability functions are defined as follows:

$$A_{11}(t) = \begin{cases} 0 & \text{if } 0 \le t < 4,\ 6 \le t < 9 \text{ or } 14 \le t < 18 \\ 1 & \text{otherwise} \end{cases} \tag{3.16}$$

$$A_{21}(t) = \begin{cases} 0 & \text{if } 0 \le t < 2,\ 4 \le t < 8,\ 9 \le t < 11 \text{ or } 13 \le t < 17 \\ 1 & \text{otherwise} \end{cases} \tag{3.17}$$

$$A_{12}(t) = \begin{cases} 0 & \text{if } 2 \le t < 3,\ 6 \le t < 10 \text{ or } 14 \le t < 16 \\ 1 & \text{otherwise} \end{cases} \tag{3.18}$$

$$A_{22}(t) = \begin{cases} 0 & \text{if } 1 \le t < 5,\ 7 \le t < 9 \text{ or } 11 \le t < 15 \\ 1 & \text{otherwise} \end{cases} \tag{3.19}$$

$$(3.20) \quad A_{32}(t) = \begin{cases} 0 & \text{if } 0 \le t < 4,\ 5 \le t < 8 \text{ or } 15 \le t < 18 \\ 1 & \text{otherwise} \end{cases}$$

$$(3.21) \quad A_{13}(t) = \begin{cases} 0 & \text{if } 0 \le t < 2,\ 7 \le t < 9,\ 14 \le t < 16 \text{ or } 17 \le t < 18 \\ 1 & \text{otherwise} \end{cases}$$

$$(3.22) \quad A_{14}(t) = \begin{cases} 0 & \text{if } 0 \le t < 2,\ 5 \le t < 8 \text{ or } 10 \le t < 16 \\ 1 & \text{otherwise} \end{cases}$$

$$(3.23) \quad A_{24}(t) = \begin{cases} 0 & \text{if } 1 \le t < 3,\ 6 \le t < 10 \text{ or } 15 \le t < 18 \\ 1 & \text{otherwise} \end{cases}$$

$$(3.24) \quad A_{34}(t) = \begin{cases} 0 & \text{if } 2 \le t < 6,\ 8 \le t < 9 \text{ or } 14 \le t < 16 \\ 1 & \text{otherwise} \end{cases}$$

$$(3.25) \quad A_{15}(t) = \begin{cases} 0 & \text{if } 4 \le t < 8 \text{ or } 14 \le t < 18 \\ 1 & \text{otherwise} \end{cases}$$

$$(3.26) \quad A_{25}(t) = \begin{cases} 0 & \text{if } 0 \le t < 5,\ 8 \le t < 14 \text{ or } 16 \le t < 17 \\ 1 & \text{otherwise} \end{cases}$$

Figure 3.3 provides a graphical illustration of equipment usage based on these equipment availability functions. Now, suppose we are given a test request of the form (1, 2, 1, 2, 1; 4) and it is desired to schedule, if possible, this test configuration at time S = 10. For k = 1, ..., 5 the following observations can be made:

<u>Case: k = 1</u>

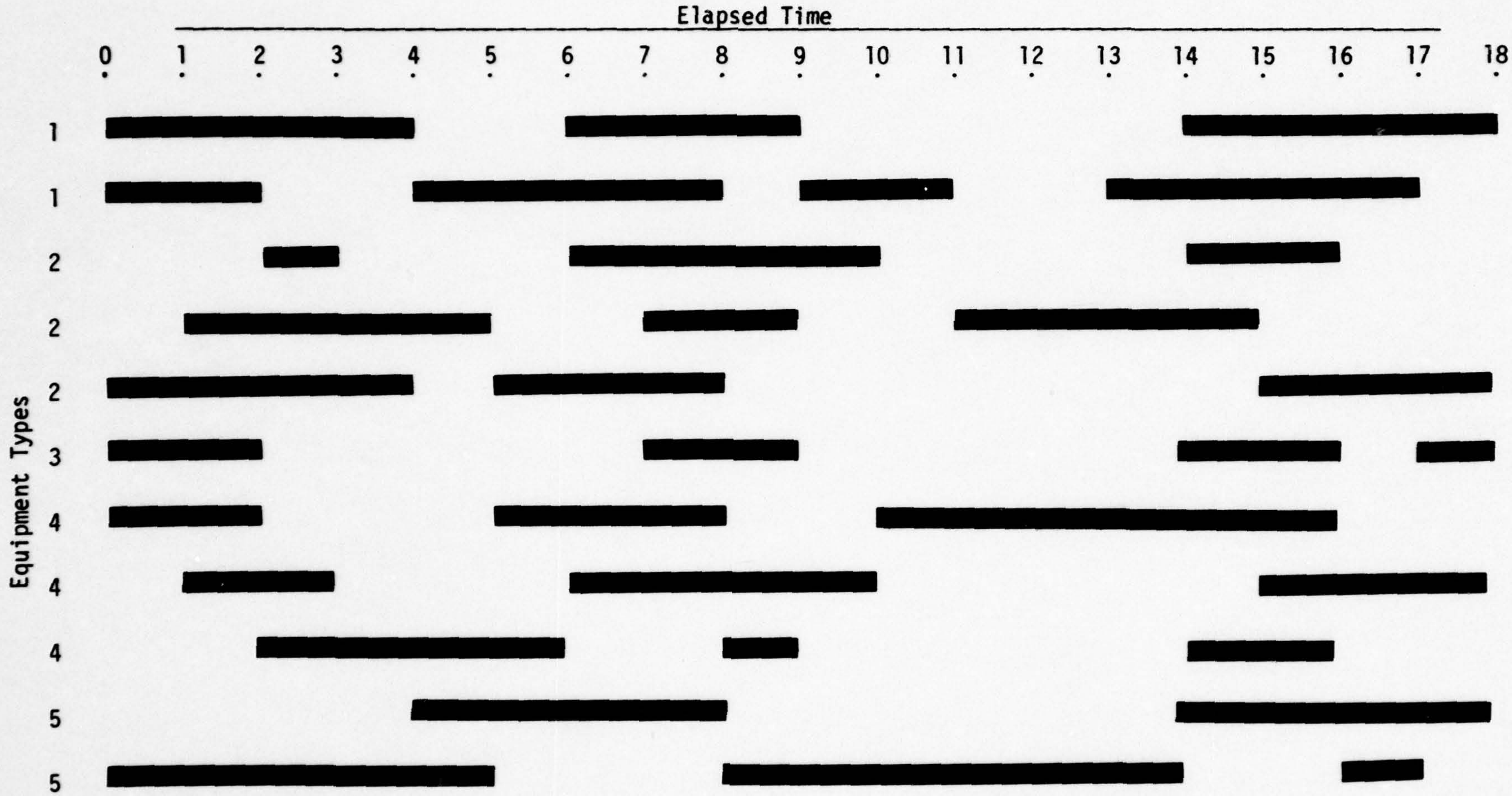
$J_1 = \{1\}$

$C_1 = 1$

$I_1 = \{1\}$

$\sum_{t=10}^{13} A_{11}(t) - 4 = 0$; thus, using the first equipment of type 1, the configuration is schedulable relative to equipments of type 1.

Figure 3.3 Illustrative Example of Equipment Usage Schedule*



*Solid lines denote equipment being used.

Case: $k = 2$

$J_2 = \{1, 2, 3\}$

$C_2 = \binom{3}{2} = 3$

$I_1 = \{1, 2\} \quad I_2 = \{1, 3\} \quad I_3 = \{2, 3\}$

$$\sum_{t=10}^{13} A_{v2}(t) - 4 = \begin{cases} 0 & \text{for } v = 1 \text{ or } 3 \\ -3 & \text{for } v = 2 \end{cases}$$

thus, using the first and third equipments of type 2, the configuration is schedulable relative to equipments of type 2.

Case: $k = 3$

$J_3 = \{1\}$

$C_3 = 1$

$I_1 = \{1\}$

$\sum_{t=10}^{13} A_{13}(t) - 4 = 0$; thus the configuration is schedulable relative to equipments of type 3.

Case: $k = 4$

$J_4 = \{2, 3\}$

$C_4 = 1$

$I_1 = \{2, 3\}$

$\sum_{t=10}^{13} A_{v4}(t) - 4 = 0$ for $v = 2$ or 3; thus, using the second and third equipment of type 4, the configuration is schedulable relative to equipments of type 4.

Case: $k = 5$

$J_5 = \{1\}$

$C_5 = 1$

$I_1 = \{1\}$

$\sum_{t=10}^{13} A_{15}(t) - 4 = 0$; thus, using the first equipment of type 5, the configuration is schedulable relative to equipments of type 5.

We conclude then that the configuration (1, 2, 1, 2, 1; 4) is schedulable at time S = 10; however, if the desired test time was 5 time units rather than 4 time units, then the earliest it could be scheduled would be at time 18, assuming the required equipments were all available for the next 5 time units thereafter.

Appendix C presents a computer listing (in Fortran IV) of the preceding schedulability procedure, and Appendix D presents an illustrative example using this computer program.

4.0 SCHEDULING ALGORITHM

The formulation of the EDATL scheduling problem was previously presented and discussed in Section 2.0. In this section we present the solution algorithm for solving the scheduling problem.

Suppose we are given C test requests of the form $(x_{11}, x_{12}, \ldots, x_{1N}; T_1, S_1, E_1)$, $(x_{21}, x_{22}, \ldots, x_{2N}; T_2, S_2, E_2)$, ..., $(x_{C1}, x_{C2}, \ldots, x_{CN}; T_C, S_C, E_C)$ and consider the time interval $[0,T]$. For $t = 0, 1, \ldots, T$ we define

ELIG(t) = set of all those configurations which are eligible to be scheduled at time t

and Problem t to mean

$$\text{maximize} \sum_{i \in ELIG(t)} f_i(t)\, y_i \tag{4.1}$$

subject to

$$\sum_{j=1}^{N_k} z_{ijk} = x_{ik}\, y_i \qquad \text{for } i \text{ in } ELIG(t) \text{ and } k=1, \ldots, N \tag{4.2}$$

$$z_{ijk}\left(\sum_{x=t}^{t+T_i-1} A_{jk}(x) - T_i\right) = 0 \qquad \text{for } i \text{ in } ELIG(t),\ j=1, \ldots, N_k \text{ and } k=1, \ldots, N \tag{4.3}$$

$$\sum_{i \in ELIG(t)} z_{ijk} \leq A_{jk}(t) \qquad \text{for } j=1, \ldots, N_k \text{ and } k=1, \ldots, N \tag{4.4}$$

$$z_{ijk},\ y_i = 0 \text{ or } 1 \qquad \text{for } i \text{ in } ELIG(t),\ j=1, \ldots, N_k \text{ and } k=1, \ldots, N \tag{4.5}$$

Initially, at t=0 we would have

$$ELIG(0) = \left\{ i : i=1, \ldots, C \text{ and } S_i = 0 \right\}. \tag{4.6}$$

If $ELIG(0) = \emptyset$, then the smallest value of t for which $ELIG(t) \neq \emptyset$ would be given by $t = \min_{i=1,\ldots,C} S_i$; hence this would be the normal starting point for the derivation of the optimal schedule.

Basically, the solution approach consists of solving a sequence of 0-1 linear programming problems until all requests have been scheduled (or as many as possible within the selected time interval [0,T]). Figure 4.1 presents an overview of the scheduling procedure. The key step in this procedure is the computation of the set ELIG(t). In general, ELIG(t) will include the $i^{\underline{th}}$ configuration if $S_i = t$ or $S_i \geq t$ and $t + T_i \leq E_i$; however, since it is not necessarily true that ELIG(t-1) = $\{i : y_i = 1$ in the solution of Problem t-1$\}$, ELIG(t) must also include those configurations which were eligible for scheduling at time t-1 but were not selected in the solution to Problem t-1. Therefore, at each step in the procedure ELIG(t) must be computed on the basis of giving consideration to those configurations which are eligible to be tested at time t and to those configurations which were eligible to be scheduled earlier but were not part of the optimal solutions computed previously.

To illustrate this procedure, consider the following example consisting of 6 equipment types with 2 equipments of type 1 ($N_1 = 2$), 3 equipments of type 2 ($N_2 = 3$), 2 equipments of type 3 ($N_3 = 2$), 3 equipments of type 4 ($N_4 = 3$), 4 equipments of type 5 ($N_5 = 4$), and 1 equipment of type 6 ($N_6 = 1$). Assume that the scheduling period of interest is [0,20] and the initial values of the equipment availability functions are given by Table 4.1. Suppose that the test configurations to be scheduled are as follows:

Number	Test Configuration
1	(1,2,1,1,2,0;5,0,8)
2	(0,2,2,2,1,1;3,0,0)
3	(1,1,0,1,2,1;4,0,4)
4	(0,0,1,1,2,1;3,3,20)
5	(1,0,0,0,1,0;7,3,16)
6	(0,1,1,2,0,0;2,6,10)
7	(1,1,1,0,0,0;2,6,16)
8	(0,1,1,0,0,1;4,6,13)
9	(1,1,0,0,0,0;5,7,12)
10	(0,1,1,2,1,0;3,8,13)

Figure 4.1 Overview of the Scheduling Procedure

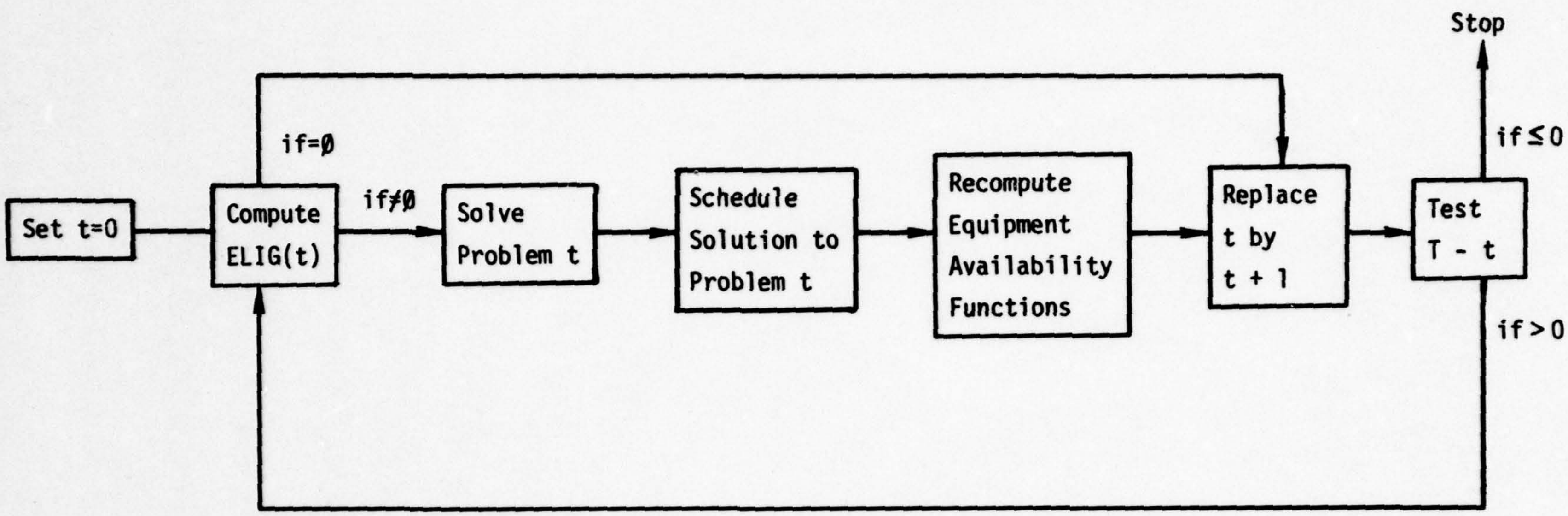


Table 4.1 Equipment Availability Function Values at Step 1*

Equipment Number (j)	Equipment Type (k)	Time 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
3	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	5	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1
3	5	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1
4	5	1	1	1	1	1	1	1	0	0	0	1	1	1	1	0	0
1	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

*0's denote blockage because of prior commitment.

The criterion for scheduling in this example will be based on maximizing the revenue derived from equipment usage where

R_k = revenue per time unit for usage of equipments of type k

$$= \begin{cases} 1 & \text{for } k = 1, 2 \\ 2 & \text{for } k = 3, 4 \\ 3 & \text{for } k = 5, 6 \end{cases} \tag{4.7}$$

Using (4.7), it is easily shown that the revenue per test configuration is given by the following:

Configuration Number (i)	Revenue ($f_i(t)$)
1	13
2	16
3	11
4	13
5	4
6	7
7	4
8	6
9	2
10	10

Following the procedure outlined in Figure 4.1, the steps are as follows:

Step 1

$$\text{ELIG}(0) = \{1, 2, 3\} \tag{4.8}$$

The Problem 0 solution is given by

$$y_i = \begin{cases} 1 & \text{if } i = 1, 3 \\ 0 & \text{if } i = 2 \end{cases} \tag{4.9}$$

$$z_{ijk} = \begin{cases} 1 & \text{for } (1,2,1), (1,1,2), (1,2,2), (1,1,3), (1,1,4), (1,1,5), \\ & (1,4,5), (3,1,1), (3,3,2), (3,2,4), (3,2,5), (3,3,5), \\ & (3,1,6) \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

Thus, configurations 1 and 3 are scheduled at time $t = 0$, however configuration 2 is unable to be scheduled at its desired start time.

Step 2

Referring to Table 4.2, we observe that no scheduling of test requests is possible at time $t = 1, 2, 3$, or 4; thus, consider $t = 5$.

(4.11) $$\mathrm{ELIG}(5) = \{2, 4, 5\}$$

The Problem 5 solution is given by

(4.12) $$y_i = \begin{cases} 1 & \text{if } i = 2 \\ 0 & \text{if } i = 4, 5 \end{cases}$$

(4.13) $$z_{ijk} = \begin{cases} 1 & \text{for } (2,1,2), (2,3,2), (2,1,3), (2,2,3), (2,1,4), \\ & \quad (2,2,4), (2,1,5), (2,1,6) \\ 0 & \text{otherwise} \end{cases}$$

Thus, it is only possible to schedule configuration 2.

Step 3

Referring to Table 4.3, we observe that no scheduling of test requests is possible at time $t = 6$; thus, consider $t = 7$.

(4.14) $$\mathrm{ELIG}(7) = \{4, 5, 6, 7, 8, 9\}$$

The Problem 7 solution is given by

(4.15) $$y_i = \begin{cases} 1 & \text{if } i = 9 \\ 0 & \text{if } i = 4, 5, 6, 7, 8 \end{cases}$$

(4.16) $$z_{ijk} = \begin{cases} 1 & \text{for } (9,2,1), (9,2,2) \\ 0 & \text{otherwise} \end{cases}$$

Thus, it is only possible to schedule configuration 9.

Step 4

(4.17) $$\mathrm{ELIG}(8) = \{4, 5, 6, 7, 8, 10\}$$

The Problem 8 solution is given by

(4.18) $$y_i = \begin{cases} 1 & \text{if } i = 5, 6, 8 \\ 0 & \text{if } i = 4, 7, 10 \end{cases}$$

Table 4.2 Equipment Availability Function Values at Step 2

Equipment Number (j)	Equipment Type (k)	Time 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	1	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
1	2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2	2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
3	2	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	3	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	4	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2	4	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
3	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	5	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
3	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
4	5	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0
1	6	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Table 4.3 Equipment Availability Function Values at Step 3

Equipment Number (j)	Equipment Type (k)	Time 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	1	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
1	2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
3	2	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
1	3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	3	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
1	4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	4	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
3	5	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
4	5	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0
1	6	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1

$$(4.19)\quad z_{ijk} = \begin{cases} 1 & \text{for } (5,1,1),\ (5,1,5),\ (6,1,2),\ (6,1,3),\ (6,1,4), \\ & \quad (6,3,4),\ (8,3,2),\ (8,2,3),\ (8,1,6) \\ 0 & \text{otherwise} \end{cases}$$

Step 5

Referring to Table 4.4, we observe that no scheduling of test requests is possible at time $t = 9$; thus, consider $t = 10$.

$$(4.20)\quad \mathrm{ELIG}(10) = \{4,\ 7,\ 10\}$$

The Problem 10 solution is given by

$$(4.21)\quad y_i = \begin{cases} 1 & \text{if } i = 10 \\ 0 & \text{if } i = 4,\ 7 \end{cases}$$

$$(4.22)\quad z_{ijk} = \begin{cases} 1 & \text{for } (10,1,2),\ (10,1,3),\ (10,1,4),\ (10,3,4), \\ & \quad (10,2,5) \\ 0 & \text{otherwise} \end{cases}$$

Thus, it is only possible to schedule configuration 10.

Step 6

Referring to Table 4.5, we observe that no scheduling of test requests is possible at times $t = 11$ or 12; thus, consider $t = 13$.

$$(4.23)\quad \mathrm{ELIG}(13) = \{4,\ 7\}$$

The Problem 13 solution is given by

$$(4.24)\quad y_i = \begin{cases} 1 & \text{if } i = 4 \\ 0 & \text{if } i = 7 \end{cases}$$

$$(4.25)\quad z_{ijk} = \begin{cases} 1 & \text{for } (4,1,3),\ (4,1,4),\ (4,2,5),\ (4,3,5), \\ & \quad (4,1,6) \\ 0 & \text{otherwise} \end{cases}$$

Thus, it is only possible to schedule configuration 4.

Step 7

Referring to Table 4.6, we observe that no scheduling of the remaining test configuration, number 7, is possible until time 16 since this is when an equipment of type 3 becomes available.

Table 4.4 Equipment Availability Function Values at Step 5

Equipment Number (j)	Equipment Type (k)	Time 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
2	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
3	2	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1
1	3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
2	3	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0
1	4	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
2	4	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	4	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
3	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
4	5	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0
1	6	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1

Table 4.5 Equipment Availability Function Values at Step 6

Equipment Number (j)	Equipment Type (k)	Time 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
2	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
3	2	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1
1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
2	3	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0
1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
2	4	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	4	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	5	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1
3	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
4	5	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0
1	6	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1

Table 4.6 Equipment Availability Function Values at Step 7

Equipment Number (j)	Equipment Type (k)	Time 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
2	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
3	2	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1
1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	3	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0
1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	4	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
3	5	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
4	5	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0
1	6	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Figure 4.2 presents the resulting schedule of test configurations, showing that all can be scheduled within the test period [0,20]. It is of interest, however, to compare the desired start and completion times of these configurations with those generated by the scheduling algorithm. This comparison is as follows:

Configuration Number	Start Time		Completion Time	
	Desired	Actual	Desired	Actual
1	0	0	8	5
2	0	5	9	8
3	0	0	4	4
4	3	13	20	16
5	3	8	16	15
6	6	8	10	10
7	6	16	16	18
8	6	8	13	12
9	7	7	12	12
10	8	10	13	13

As might be expected, not all configurations were selected for scheduling at their desired start time (this only happened for 3 out of the 10 configurations) and one configuration (#7) did not have its testing completed at the desired time.

The key steps in the solution procedure, as outlined in Figure 4.1, are concerned with (1) determining ELIG(t), (2) using a 0-1 linear programming routine, (3) updating the equipment availability functions, and (4) repeating (1)–(3) until a schedule is obtained.

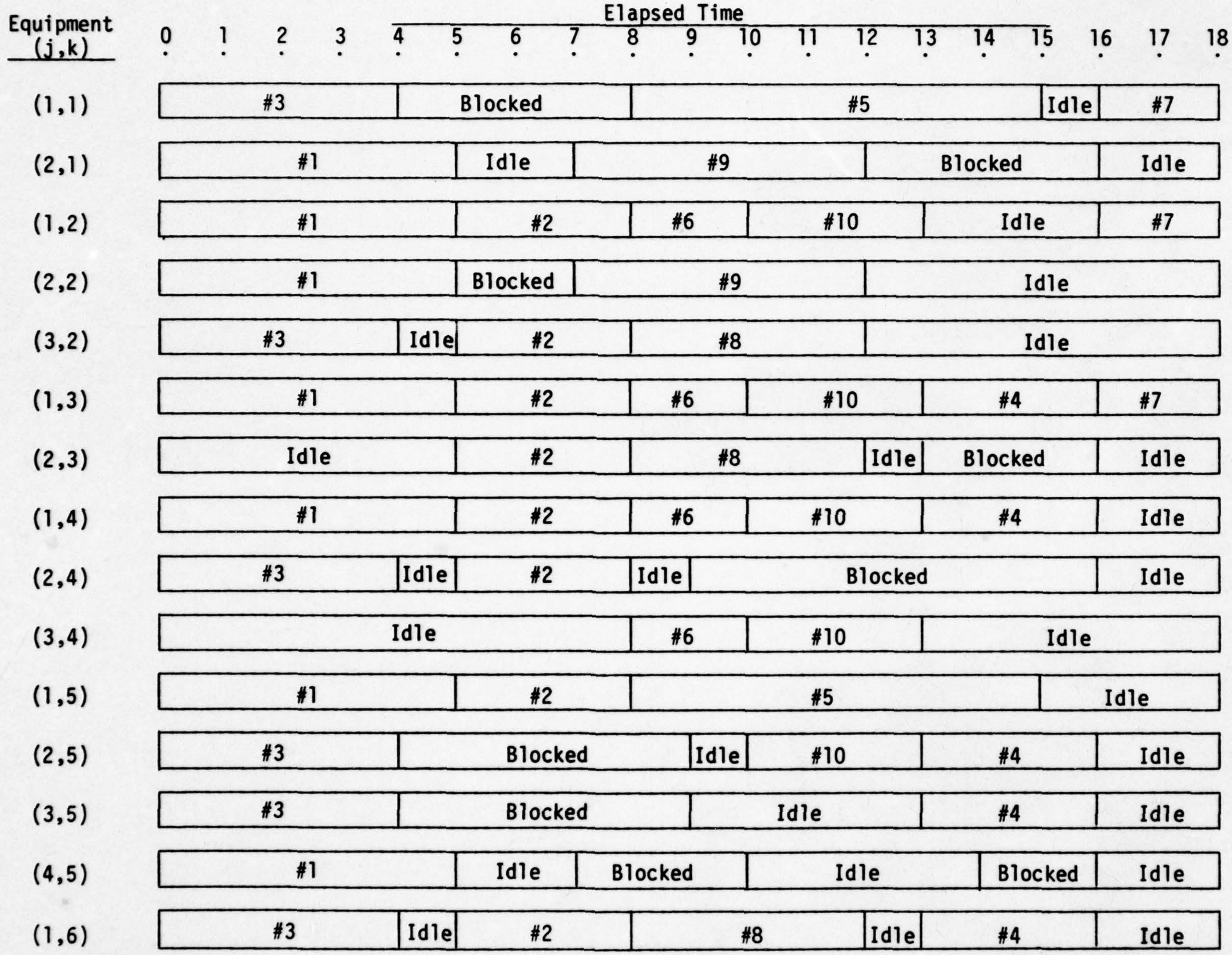
As was demonstrated in the preceding example, the general procedure for computing ELIG(t) can be described as follows:

(1) Set $t = \min_{i=1,\ldots,C} S_i$, i.e., the starting point for the derivation of the optimal schedule.

(2) Compute $\mathrm{ELIG}(t) = \left\{ i : S_i = t \right\}$.

(3) Solve Problem t.

Figure 4.2 Graphical Solution to Example Scheduling Problem



(4) Compute $SCH(t) = \left\{ \begin{array}{l} i : i \in ELIG(t) \text{ and } y_i = 1 \\ \text{in the solution to Problem } t \end{array} \right\}$

(5) Compute $C(t) = \text{Card} \bigcup_{j=t_0}^{t} SCH(j)$ where $t_0 = \min_{i=1,\ldots,c} S_i$.

(6) Test C-c(t). If = 0, scheduling is complete; otherwise, go to (7).

(7) Compute $UNSCH(t) = \left\{ \begin{array}{l} i : i \in ELIG(t) \text{ and } y_i = 0 \\ \text{in the solution to Problem } t \end{array} \right\}$

(8) Replace t by t + 1.

(9) Test T - t. If ≥ 0, go to (10); if <0, no schedule is possible in [0,T].

(10) Compute $ELIG(t) = \left\{ i : S_i = t \text{ or } i \in UNSCH(t-1) \right\}$.

(11) Test ELIG(t). If empty, go to (8), otherwise, go to (3).

With regard to the use of a 0-1 linear programming scheme, one could incorporate it as a subroutine in a more general program in which the basic steps of Figure 4.1 are followed, or one could use a pseudo-automated scheme in which ELIG(t) is computed manually, then input to the computerized 0-1 linear programming package, the equipment availability functions are then recomputed manually given the solution, then ELIG(t+1) is computed manually, and, finally, this process is repeated until a schedule is obtained.

The updating of the equipment availability functions is a straight forward process. If $z_{ijk} = 1$ in the solution to Problem t, then we merely redefine $A_{jk}(x) = 0$ for $x = t, t+1, \ldots, t+T_i-1$. On the other hand, if $z_{ijk} = 0$ in the solution to Problem t, then no redefinition of the $A_{jk}(x)$'s is necessary.

Finally, this overall process is repeated until the schedule is obtained.

BIBLIOGRAPHY

(1) First Preliminary Report on the EDATL Facility Scheduling Problem Research, Ultrasystems, Inc., 22 September 1976

(2) Second Preliminary Report on the EDATL Facility Scheduling Problem Research, Ultrasystems, Inc., 4 April 1977

(3) Third Report on the EDATL Facility Scheduling Problem Research, Ultrasystems, Inc., 9 January 1978

(4) EDATL Program Management Plan, Naval Electronics Laboratory Center, 8 September 1975

(5) Giffler, B. and G. L. Thompson, "Algorithms for Solving Production – Scheduling Problems," Oper. Res. (1960), pp 487-503

(6) Thesen, A., "Heuristic Scheduling of Activities Under Resource and Precedence Restrictions," Man. Sci., Vol. 23 No. 4, December 1976, pp 412-422

(7) Babad, J. M., V. Balachandran and E. A. Stohr, "Management of Program Storage in Computers," Man. Sci., Vol. 23 No. 4, December 1976, pp 380-390

(8) Panwalkar, S. S. and W. Iskander, "A Survey of Scheduling Rules," Oper. Res. Vol. 25 No. 1, January-February 1977, pp 45-61

(9) "Description of Test Scheduling Algorithm, Eglin AFB, Florida," Transmitted on 6 October 1976 to Ultrasystems by Directorate of Computer Sciences, Headquarters Armament Development and Test Center, Eglin AFB

APPENDIX A

Computer Program Listing For
Schedule-According-To-Decreasing-Time-
Requirements Method and Schedule-As-
Soon-As-Possible Method

```
      PROGRAM SCHDL (INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)
      COMMON N,NC,NE(20),X(20,20),T(20),IND(20),ST(20),ET(20),TMAX
      INTEGER X,T,ST,ET,TMAX
      NAMELIST/VAR/N,NC,NE,X,T
      READ VAR
      PRINT VAR
      CALL SADTR
      CALL SASAP
      STOP
      END
      SUBROUTINE SADTR
      DIMENSION S(20,20),LTIME(20),ISTAR(20)
      COMMON N,NC,NE(20),X(20,20),T(20),IND(20),ST(20),ET(20),TMAX
      INTEGER X,T,ST,ET,TMAX
      INTEGER S
      CALL ORDTIM
      NS=1
      J=1
      S(IND(1),1)=1
      ISTAR(1)=0
      ET(IND(1))=T(IND(1))
      LTIME(1)=T(IND(1))
      ST(IND(1))=0
      DO 6 I=2,NC
    1 CONTINUE
      DO 3 K=1,N
      ISUM=0
      DO 2 L=1,NC
      ISUM=ISUM+S(IND(L),J)*X(IND(L),K)
    2 CONTINUE
      NBC=ISUM+X(IND(I),K)
      IF (NBC.GT.NE(K)) GO TO 4
    3 CONTINUE
      S(IND(I),J)=1
      ST(IND(I))=ISTAR(J)
      ET(IND(I))=ISTAR(J)+T(IND(I))
      GO TO 5
    4 CONTINUE
      J=J+1
      IF (J.LE.NS) GO TO 1
      NS=NS+1
      S(IND(I),NS)=1
      ISTAR(NS)=LTIME(NS-1)
      ST(IND(I))=LTIME(NS-1)
      ET(IND(I))=LTIME(NS-1)+T(IND(I))
      LTIME(NS)=ET(IND(I))
      J=1
    5 CONTINUE
    6 CONTINUE
      TMAX=LTIME(NS)
      PRINT 11
   11 FORMAT(1H1,10X*START/END TIMES*)
      DO 9 I=1,NC
```

```
      DO 9 L=1,NS
      IF (S(I,L).EQ.1) PRINT 8, IND(I),  ST(IND(I)),ET(IND(I))
    8 FORMAT(//,10X*REQUEST*1X,I2,8X*START TIME IS*1X,I2,5X*END TIME IS*
     *     ,1X,I2)
    9 CONTINUE
      PRINT 10, TMAX
   10 FORMAT(////,10X*MAXIMUM TIME TO SCHEDULE IS*1X,I2)
      CALL OUTFOR
      RETURN
      END
      SUBROUTINE ORDTIM
      COMMON N,NC,NE(20),X(20,20),T(20),IND(20),ST(20),ET(20),TMAX
      INTEGER X,T,ST,ET,TMAX
      DO 1 I=1,NC
      IND(I)=I
    1 CONTINUE
      JQ=NC-1
      DO 4 I=1,JQ
      J=I
    2 CONTINUE
      IF (T(IND(J)).GE.T(IND(J+1))) GO TO 3
      NBC=IND(J)
      IND(J)=IND(J+1)
      IND(J+1)=NBC
      IF (J.EQ.1) GO TO 3
      J=J-1
      GO TO 2
    3 CONTINUE
    4 CONTINUE
      RETURN
      END
      SUBROUTINE SASAP
      DIMENSION Q(20,20),TNOW(20)
      COMMON N,NC,NE(20),X(20,20),T(20),IND(20),ST(20),ET(20),TMAX
      INTEGER X,T,ST,ET,TMAX
      INTEGER Q,TNOW
      CALL ORDTIM
      DO 19 I=1,NC
      ST(I)=0
      ET(I)=0
   19 CONTINUE
      I=1
      J=1
      TNOW(1)=0
      ST(IND(1))=0
      ET(IND(1))=T(IND(I))
      Q(IND(I),J)=1
    1 CONTINUE
      I=I+1
      IF (NC-I.LT.0) GO TO 5
    2 CONTINUE
      DO 4 K=1,N
      ISUM=0
      DO 3 L=1,NC
      ISUM=ISUM+X(IND(L),K)*Q(IND(L),J)
    3 CONTINUE
      NBC=ISUM+X(IND(I),K)
      IF (NBC.GT.NE(K)) GO TO 1
    4 CONTINUE
      Q(IND(I),J)=1
```

```
      ST(IND(I))=TNOW(J)
      ET(IND(I))=TNOW(J)+T(IND(I))
      GO TO 1
    5 CONTINUE
      DO 21 IB=1,NC
      ISUM=0
      DO 6 IX=1,J
      ISUM=ISUM+Q(IND(IB),IX)
    6 CONTINUE
      IF (ISUM.EQ.0) GO TO 22
   21 CONTINUE
      GO TO 13
   22 CONTINUE
      J=J+1
      IXX=0
      DO 7 IQ=1,NC
      IXX=IXX+T(IND(IQ))
    7 CONTINUE
      TNOW(J)=IXX
      DO 8 IP=1,NC
      IF (TNOW(J-1).LT.ET(IND(IP))) TNOW(J)=MIN0(TNOW(J),ET(IND(IP)))
    8 CONTINUE
      JQ=J-1
      DO 9 IL=1,NC
      IF (Q(IND(IL),JQ).EQ.1.AND.TNOW(J).LT.ET(IND(IL))) Q(IND(IL),J)=1
    9 CONTINUE
      DO 12 IZ=1,NC
      JQ=J-1
      ISUM=0
      DO 10 JZ=1,JQ
      ISUM=ISUM+Q(IND(IZ),JZ)
   10 CONTINUE
      IF (ISUM.GE.1)  GO TO 11
      I=IZ
      GO TO 2
   11 CONTINUE
   12 CONTINUE
   13 CONTINUE
      TMAX=0
      DO 14 IZZ=1,NC
      IF (ET(IND(IZZ)).GT.TMAX) TMAX=ET(IND(IZZ))
   14 CONTINUE
   18 FORMAT(1H1,10X*START/END TIMES*)
      PRINT 18
      DO 16 I=1,NC
      PRINT 15, IND(I),ST(IND(I)),ET(IND(I))
   15 FORMAT(//,10X*REQUEST*1X,I2,8X*START TIME IS*1X,I2,5X*END TIME IS*
     *     ,1X,I2)
   16 CONTINUE
      PRINT 17,TMAX
   17 FORMAT(////,10X*MAXIMUM TIME TO SCHEDULE IS*1X,I2)
      CALL OUTFOR
      RETURN
      END
      SUBROUTINE ORDX
      COMMON N,NC,NE(20),X(20,20),T(20),IND(20),ST(20),ET(20),TMAX
      INTEGER X,T,ST,ET,TMAX
      DO 1 I=1,NC
      IND(I)=I
    1 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE OUTFUR
      DIMENSION U(20,5,20)
      COMMON N,NC,NE(20),X(20,20),T(20),IND(20),ST(20),ET(20),TMAX
      INTEGER X,T,ST,ET,TMAX
      INTEGER U
      DO 16 K=1,N
      JZ=NE(K)
      DO 15 JY=1,JZ
      DO 15 L=1,TMAX
      U(K,JY,L)=0
   15 CONTINUE
   16 CONTINUE
      DO 6 I=1,NC
      DO 5 K=1,N
      IF (X(IND(I),K).EQ.0) GO TO 4
      JJ=1
    1 CONTINUE
      JX=X(IND(I),K)
      JY=ST(IND(I))+1
      JZ=ET(IND(I))
      DO 7 JQ=JY,JZ
      IF (U(K,JJ,JQ).GT.0) GO TO 3
    7 CONTINUE
      JX=JJ+JX-1
      DO 2 LX=JJ,JX
      DO 2 LY=JY,JZ
      U(K,LX,LY)=IND(I)
    2 CONTINUE
      GO TO 4
    3 CONTINUE
      JJ=JJ+1
      GO TO 1
    4 CONTINUE
    5 CONTINUE
    6 CONTINUE
      IM=TMAX
      PRINT 10, (I   ,I=1,IM)
   10 FORMAT(1H1,50X*SCHEDULE*,//,5X*TIME*,13X,20(I2,3X))
      PRINT 11
   11 FORMAT(//,5X*EQUIP*,3X*ITEM*)
      DO 14 K=1,N
      JA=NE(K)
      DO 13 J=1,JA
      PRINT 12, K,J,(U(K,J,IT),IT=1,IM)
   12 FORMAT(//,7X,I2,6X,I1,6X,20(I2,3X))
   13 CONTINUE
   14 CONTINUE
      RETURN
      END
```

APPENDIX B

Illustrative Examples Using Schedulability Computer Program

BASIC DATA COMMON TO ALL EXAMPLES:

N = number of equipment types = 4

N_1 = number of equipments of type 1 = 1

N_2 = number of equipments of type 2 = 2

N_3 = number of equipments of type 3 = 3

N_4 = number of equipments of type 4 = 4

C = number of requests = 9

EXAMPLE 1 Schedule-According-To-Decreasing-Time-Requirements Method

Request Number	Request Type
1	(0,0,1,0;6)
2	(1,0,0,1;4)
3	(0,2,0,0;5)
4	(0,0,2,3;4)
5	(1,2,0,1;2)
6	(0,0,0,3;1)
7	(0,0,2,3;2)
8	(0,0,3,0;3)
9	(1,1,2,2;3)

EXAMPLE 2 Schedule-As-Soon-As-Possible Method
(Order by first-come first-served)

Request Number	Request Type
1	(0,0,1,0;6)
2	(1,0,0,1;4)
3	(0,2,0,0;5)
4	(0,0,2,3;4)

Request Number	Request Type
5	(1,2,0,1;2)
6	(0,0,0,3;1)
7	(0,0,2,3;2)
8	(0,0,3,0;3)
9	(1,1,2,2;3)

EXAMPLE 3 Schedule-As-Soon-As-Possible Method
(Order according to decreasing times)

Request Number	Request Type
1	(0,0,1,0;6)
2	(1,0,0,1;4)
3	(0,2,0,0;5)
4	(0,0,2,3;4)
5	(1,2,0,1;2)
6	(0,0,0,3;1)
7	(0,0,2,3;2)
8	(0,0,3,0;3)
9	(1,1,2,2;3)

Table B.1 Example 1 Solution

```
START/END TIMES

REQUEST  1        START TIME IS  0     END TIME IS  6

REQUEST  3        START TIME IS  0     END TIME IS  5

REQUEST  2        START TIME IS  0     END TIME IS  4

REQUEST  4        START TIME IS  0     END TIME IS  4

REQUEST  8        START TIME IS  6     END TIME IS  9

REQUEST  9        START TIME IS  9     END TIME IS 12

REQUEST  5        START TIME IS  6     END TIME IS  8

REQUEST  7        START TIME IS 12     END TIME IS 14

REQUEST  6        START TIME IS  6     END TIME IS  7


MAXIMUM TIME TO SCHEDULE IS 14
```

Table B.2 Example 1 Equipment Usage Schedule

SCHEDULE

TIME		1	2	3	4	5	6	7	8	9	10	11	12	13	14
EQUIP	ITEM														
1	1	2	2	2	2	0	0	5	5	0	9	9	9	0	0
2	1	3	3	3	3	3	0	5	5	0	9	9	9	0	0
2	2	3	3	3	3	3	0	5	5	0	0	0	0	0	0
3	1	1	1	1	1	1	1	8	8	8	9	9	9	7	7
3	2	4	4	4	4	0	0	8	8	8	9	9	9	7	7
3	3	4	4	4	4	0	0	8	8	8	0	0	0	0	0
4	1	2	2	2	2	0	0	5	5	0	9	9	9	7	7
4	2	4	4	4	4	0	0	6	0	0	9	9	9	7	7
4	3	4	4	4	4	0	0	6	0	0	0	0	0	7	7
4	4	4	4	4	4	0	0	6	0	0	0	0	0	0	0

Table B.3 Example 2 Solution

```
START/END TIMES

REQUEST  1        START TIME IS  0     END TIME IS  6

REQUEST  2        START TIME IS  0     END TIME IS  4

REQUEST  3        START TIME IS  0     END TIME IS  5

REQUEST  4        START TIME IS  0     END TIME IS  4

REQUEST  5        START TIME IS  5     END TIME IS  7

REQUEST  6        START TIME IS  5     END TIME IS  6

REQUEST  7        START TIME IS  6     END TIME IS  8

REQUEST  8        START TIME IS  8     END TIME IS 11

REQUEST  9        START TIME IS 11     END TIME IS 14


MAXIMUM TIME TO SCHEDULE IS 14
```

Table B.4 Example 2 Equipment Usage Schedule

		SCHEDULE													
TIME		1	2	3	4	5	6	7	8	9	10	11	12	13	14
EQUIP	ITEM														
1	1	2	2	2	2	0	5	5	0	0	0	0	9	9	9
2	1	3	3	3	3	3	5	5	0	0	0	0	9	9	9
2	2	3	3	3	3	3	5	5	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	7	7	8	8	8	9	9	9
3	2	4	4	4	4	0	0	7	7	8	8	8	9	9	9
3	3	4	4	4	4	0	0	0	0	8	8	8	0	0	0
4	1	2	2	2	2	0	5	5	0	0	0	0	9	9	9
4	2	4	4	4	4	0	6	7	7	0	0	0	9	9	9
4	3	4	4	4	4	0	6	7	7	0	0	0	0	0	0
4	4	4	4	4	4	0	6	7	7	0	0	0	0	0	0

Table B.5 Example 3 Solution

```
START/END TIMES

REQUEST  1        START TIME IS  0     END TIME IS  6

REQUEST  3        START TIME IS  0     END TIME IS  5

REQUEST  2        START TIME IS  0     END TIME IS  4

REQUEST  4        START TIME IS  0     END TIME IS  4

REQUEST  8        START TIME IS  6     END TIME IS  9

REQUEST  9        START TIME IS  9     END TIME IS 12

REQUEST  5        START TIME IS  7     END TIME IS  9

REQUEST  7        START TIME IS  4     END TIME IS  6

REQUEST  6        START TIME IS  8     END TIME IS  9


MAXIMUM TIME TO SCHEDULE IS 12
```

Table B.6 Example 3 Equipment Usage Schedule

SCHEDULE

TIME		1	2	3	4	5	6	7	8	9	10	11	12
EQUIP	ITEM												
1	1	2	2	2	2	0	0	0	5	5	9	9	9
2	1	3	3	3	3	3	0	0	5	5	9	9	9
2	2	3	3	3	3	3	0	0	5	5	0	0	0
3	1	1	1	1	1	1	1	8	8	8	9	9	9
3	2	4	4	4	4	7	7	8	8	8	9	9	9
3	3	4	4	4	4	7	7	8	8	8	0	0	0
4	1	2	2	2	2	7	7	0	5	5	9	9	9
4	2	4	4	4	4	7	7	0	0	6	9	9	9
4	3	4	4	4	4	7	7	0	0	6	0	0	0
4	4	4	4	4	4	0	0	0	0	6	0	0	0

APPENDIX C

Computer Program Listing For
Fitting of Test Request Into
Existing Schedule

```
      PROGRAM NUSCHD (INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)
      DIMENSION X(20),A(5,20,20),NE(20),U(20,5)
      INTEGER X,A,T,U
      NAMELIST/VAR/ X,T,A,N,NE,NEST,NMST
      READ VAR
      PRINT VAR
      PRINT 39
   39 FORMAT(1H1,5X*SCHEDULABILITY OF NEW REQUEST*)
      DO 11 NS=NEST,NMST
      DO 21 K=1,N
      JX=NE(K)
      DO 20 J=1,JX
      U(K,J)=0
   20 CONTINUE
   21 CONTINUE
      DO 6 K=1,N
      IF (X(K).EQ.0) GO TO 5
      ISUM=0
      JQ=NE(K)
      DO 1 J=1,JQ
      ISUM=ISUM+A(J,K,NS+1)
    1 CONTINUE
      IF (ISUM.LT.X(K)) GO TO 8
      INDEX=0
      DO 4 J=1,JQ
      NA=NS+T-1
      DO 2 L=NS,NA
      IF (A(J,K,L+1).EQ.0) GO TO 3
    2 CONTINUE
      INDEX=INDEX+1
      U(K,J)=1
      IF (INDEX.EQ.X(K)) GO TO 5
    3 CONTINUE
    4 CONTINUE
      GO TO 8
    5 CONTINUE
    6 CONTINUE
      PRINT 7
    7 FORMAT(     //,10X*NEW REQUEST IS SCHEDULABLE*)
      IEND=NS+T
      PRINT 30, NS,IEND
   30 FORMAT(///,10X*START TIME IS*,1X,I2,10X*END TIME IS*,1X,I2)
      PRINT 31
   31 FORMAT(///,15X*EQUIPMENT ASSIGNMENT*,/,20X*EQUIPMENT*,5X,
     *     *TYPE*,5X*ASSIGNMENT*)
      DO 34 K=1,N
      JA=NE(K)
      DO 33 J=1,JA
      PRINT 32, K,J,U(K,J)
   32 FORMAT(/,23X,I2,10X,I1,11X,I1)
   33 CONTINUE
   34 CONTINUE
      GO TO 10
    8 CONTINUE
      PRINT9,NS
    9 FORMAT(//,10X*NEW REQUEST IS NOT SCHEDUABLE AT START TIME*,
     *     1X,I2)
   10 CONTINUE
   11 CONTINUE
      STOP
      END
```

APPENDIX D

Illustrative Example Using Computer Program
To Fit Given Test Request Into
Existing Schedule

Test Request = (1,1,1,1;2)

Earliest Desired Start Time = 2

Latest Desired Start Time = 12

Initial equipment availability matrix is as follows:

		Time											
Equipment	Item	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	0	1	1	0	0	1	0	0	0	1	1
2	1	0	0	0	1	0	0	1	0	0	0	1	1
2	2	0	0	0	1	0	0	1	1	1	1	1	1
3	1	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	1	1	0	0	0	0	0	0	0	0
3	3	0	0	1	1	0	0	0	1	1	1	1	1
4	1	0	0	1	1	0	0	1	0	0	0	0	0
4	2	0	0	1	1	0	1	1	0	0	0	0	0
4	3	0	0	1	1	0	1	1	1	1	1	0	0
4	4	0	0	1	1	0	1	1	1	1	1	1	1

Table D.1 Possible Schedules for Test Request

```
SCHEDULABILITY OF NEW REQUEST

     NEW REQUEST IS NOT SCHEDUABLE AT START TIME  2

     NEW REQUEST IS NOT SCHEDUABLE AT START TIME  3

     NEW REQUEST IS SCHEDULABLE

     START TIME IS  4          END TIME IS  6

          EQUIPMENT ASSIGNMENT
```

EQUIPMENT	TYPE	ASSIGNMENT
1	1	1
2	1	0
2	2	0
3	1	0
3	2	1
3	3	0
4	1	1
4	2	0
4	3	0
4	4	0

```
     NEW REQUEST IS NOT SCHEDUABLE AT START TIME  5

     NEW REQUEST IS NOT SCHEDUABLE AT START TIME  6

     NEW REQUEST IS NOT SCHEDUABLE AT START TIME  7

     NEW REQUEST IS NOT SCHEDUABLE AT START TIME  8
```

Table C.1 (Continued)

NEW REQUEST IS NOT SCHEDUABLE AT START TIME 9

NEW REQUEST IS NOT SCHEDUABLE AT START TIME 10

NEW REQUEST IS NOT SCHEDUABLE AT START TIME 11

NEW REQUEST IS SCHEDULABLE

START TIME IS 12   END TIME IS 14

EQUIPMENT ASSIGNMENT

EQUIPMENT	TYPE	ASSIGNMENT
1	1	1
2	1	0
2	2	0
3	1	0
3	2	0
3	3	1
4	1	0
4	2	0
4	3	0
4	4	1

NEW REQUEST IS NOT SCHEDUABLE AT START TIME 13